An Application of Second Generation Wavelets for Image Denoising using Dual Tree Complex Wavelet Transform

SK.Umar Faruq[1], Dr.K.V.Ramanaiah[2], Dr.K.Soundara Rajan[3]
[1]Quba college of Engineering &Technology, Nellore, A.P, India
Email : faruq_sk2003@yahoo.co.in
[2]NBKR Institute of Science & Technology, Nellore, A.P, India.
Email :{ ramanaiahkota@gmail.com, soundararajan_jntucea@yahoo.com}

Abstract—**The lifting scheme of the discrete wavelet transform (DWT) is now quite well established as an efficient technique for image denoising. The lifting scheme factorization of biorthogonal filter banks is carried out with a linear-adaptive, delay free and faster decomposition arithmetic. This adaptive factorization is aimed to achieve a well transparent, more generalized, complexity free fast decomposition process in addition to preserve the features that an ordinary wavelet decomposition process offers. This work is targeted to get considerable reduction in computational complexity and power required for decomposition. The hard striking demerits of DWT structure viz., shift sensitivity and poor directionality had already been proven to be washed out with an emergence of dual tree complex wavelet (DT-CWT) structure. The well versed features of DT-CWT and robust lifting scheme are suitably combined to achieve an image denoising with prolific rise in computational speed and directionality, also with a desirable drop in computation time, power and complexity of algorithm compared to all other techniques.**

Index Terms— **Lifting scheme, Dual Tree Complex Wavelet transform (DT CWT), Denoising, Computational complexity, coding gain and PSNR.**

I. Introduction

One of the fundamental challenges in the field of image processing and computer vision is image denoising, where the underlying goal is to produce an estimate of the original image by suppressing noise from a noise contaminated version of the image. Image noise may be caused by different intrinsic (i.e., sensor) and extrinsic (i.e., environment) conditions which are Often not possible to avoid. Images are most frequently bear upon by noise evincing the facts like image capturing sensor internal imperfections, scarce of proper illumination of the object to be captured, during the process of its acquisition mean by digitization and its transmission. In fact the performance of the image acquiring sensors are generally affected by a wide variety of factors some among which are the factors such as environmental conditions during its acquisition process and the quality of the sensing elements deployed in image capturing systems, i.e., image acquisition sensors like Charge Coupled Device (CCD) cameras. For instance in acquiring images with a camera capturing object illumination and lighting levels and sensor temperature are major factors affecting the amount of noise in the resulting image. Another major possibility of image corruption with noise will be its transmission to a point of interest from a point of its perception by a sensor, principally due to the interference in the channel used for its transmission. Image noise is usually with reference to wide stochastic variations as opposed to deterministic distortions such as shading or lack of focus. It is actually the degree of variation of pixel values caused by the statistical nature of radioactive decay and detection processes. Even if we acquire an image of a uniform (flat) source on an ideal gamma camera with perfect uniformity and efficiency the number of counts detected in all pixels of the image will not be the same. In addition to noise added inherently by a sensor, image processing techniques also corrupt the image with noise [1]. There are a wide variety of noise models which can degrade the image, among which the most common is the additive one i.e., the additive white Gaussian noise will show a serious impact on visual perception of image by its intensive degradation. And the remaining possible noise models which have a probability of corrupting an image are salt and pepper, speckle, poison and thermal noises in the communication medium. This undesirable corruption of image by noise is inherent in any image acquisition device. Certainly it is possible to distinguish between two regimes when the degradation structure was observed by photo sensors: in the first scenario, the measured intensities are sufficiently high and the noise is assumed to be *signal-independent* degradation. In the second scenario, only a few photons are detected, leading to a strong *signal –dependent* degradation. This image degradation by noise will make the image to be null information conveying object. Removing such noise is of great benefit in many applications and this may explain vast interest in this problem and its solution. The ultimate goal of image denoising technique is to improve the degraded image in some sense by suppressing the random noise, which has corrupted the image, while preserving the most important visual features of the image, such as edges. Denoising is an essential step prior to any higher-level image-processing tasks such as segmentation, photo restoration, visual tracking where obtaining the original image content is crucial. Many algorithms have been proposed for image denoising, and there has been a fair amount of research on wavelet based image denoising, because wavelet provides an

© 2013 ACEEE
DOI: 01.IJSIP.4.3.14

appropriate basis for image denoising. But this single tree wavelet based image denoising has poor directionality, loss of phase information and shift sensitivity as its limitations. Hence we proposed dual tree wavelet structure for an improved quality image denoising while getting rid of all above limitations and lifting scheme structure for faster, complex free and efficient decomposition process.

II. Dual Tree and Lifting Frame Work

The conceptual idea central to a wavelet or sub-band transform [2],is illustrated in figure (1). The real wavelet transform makes an use of analysis filters $h_0(n)$ and $h_1(n)$,followed by sub sampling, while the reverse transform first up-samples and then uses two synthesis filters $f_0(n)$ and $f_1(n)$.Emulating same process with $\{g_0(n), g_1(n)\}$ and $\{p_0(n), p_1(n)\}$ as analysis and synthesis filter pairs[3]with suitable sub sampling and up- sampling processes will lead to an imaginary tree wavelet transform. where the analysis and synthesis filter pairs in one tree will differ with those in another tree with a half sample delay.

$$G_0(w) \simeq H_0(w)e^{-j\theta(w)};\theta(w) = w/2 \tag{1}$$

Since the filters in both trees can be made to offset by half sample, the wavelets resulting from the filter pair satisfies the Hilbert transform condition, given as

$$\Psi\ \ _i(W) \simeq \begin{pmatrix} -j\Psi_r(w); w > 0 \\ j\Psi_r(w); w < 0 \end{pmatrix} \tag{2}$$

For real and imaginary trees, the modulation Matrices [1] indicated by $M_1(z)$ and $M_2(z)$, respectively , defined as

$$M_{10} = \begin{pmatrix} F_0(z) & F_0(-z) \\ F_1(z) & F_1(-z) \end{pmatrix} \tag{3}$$

$$M_{20} = \begin{pmatrix} P_0(z) & P_0(-z) \\ P_1(z) & P_1(-z) \end{pmatrix} \tag{4}$$

The condition for perfect reconstruction can be represented in z-domain as

For real

$$\begin{gathered} H_0(-z)F_0(z) + H_1(-z)F_1(z) = 0 \\ H_0(z)F_0(z) + H_1(z)F_1(z) = 2d^{-1} \end{gathered} \tag{5}$$

For imaginary

$$\begin{gathered} G_0(-z)P_0(z) + G_1(-z)P_1(z) = 0 \\ G_0(z)P_0(z) + G_1(z)P_1(z) = 2d^{-1} \end{gathered} \tag{6}$$

All the filters are assumed to be causal for simplicity and is the cause for introducing the term z^{-d}.secondly, $M_{11}(z)$ and $M_{21}(z)$ are the dual versions of $M_{10}(z)$ and $M_{20}(z)$ respectively. We can write, for perfect reconstruction as

$$M_{11}(z^{-1})^t M_{10}(z) = 2\boldsymbol{I} \tag{7}$$

$$M_{21}(z^{-1})^t M_{20}(z) = 2\boldsymbol{I} \tag{8}$$

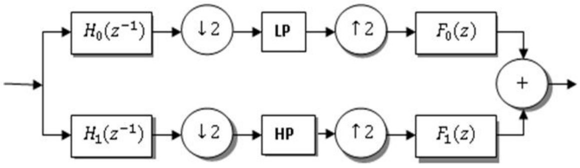


Fig 1(a): Real tree wavelet transform

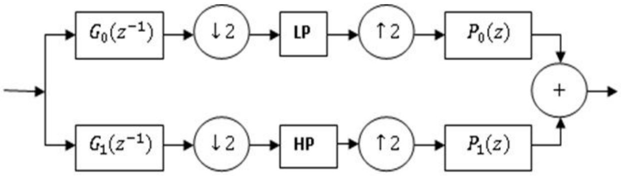


Fig 1(b): Imaginary wavelets transform.

Where **I** being an identity matrix of order '2'.If all filters are FIR, then$M_{10}(z)$, $M_{20}(z)$ and $M_{11}(z)$, $M_{21}(z)$ belongs to GL(2;R[z,z^{-1}]).As a special case in a orthogonal wavelet transform, $F_0(z) = H_0(z)$, $F_1(z) = H_1(z)$, $P_0(z) = G_0(z)$ and $P_1(z) = G_1(z)$, the modulation matrices $M_{10}(z) = M_{11}(z)$ and $M_{20}(z) = M_{21}(z)$,which are intern $\sqrt{2}$ times a unitary matrix. In poly phase notation, these filters can be written interims of their even and odd phases, according to the following relations

$$\begin{aligned} H_0(z) &= H_{0e}(z^2) + z^{-1}H_{0o}(z^2) \\ H_1(z) &= H_{1e}(z^2) + z^{-1}H_{1o}(z^2) \end{aligned} \tag{9}$$

$$\begin{aligned} G_0(z) &= G_{0e}(z^2) + z^{-1}G_{0o}(z^2 \\ G_1(z) &= G_{1e}(z^2) + z^{-1}G_{1o}(z^2) \end{aligned} \tag{10}$$

$$\begin{aligned} F_0(z) &= F_{0e}(z^2) + z^{-1}F_{0o}(z^2) \\ F_1(z) &= F_{1e}(z^2) + z^{-1}F_{1o}(z^2) \end{aligned} \tag{11}$$

$$\begin{aligned} P_0(z) &= P_{0e}(z^2) + z^{-1}P_{0o}(z^2) \\ P_1(z) &= P_{1e}(z^2) + z^{-1}P_{1o}(z^2) \end{aligned} \tag{12}$$

Where $H_{ie}(z)\ S$ contains even coefficients and $H_{io}(z)$ contains odd coefficients for i=0,1.Thus

$$\begin{aligned} H_{0e}(z) &= \sum_k h_{0_{2k}}z^{-k} \ and\ H_{0o}(z) = \sum_k h_{0_{2k+1}}z^{-k} \\ H_{1e}(z) &= \sum_k h_{1_{2k}}z^{-k} \ and\ H_{1o}(z) = \sum_k h_{1_{2k+1}}z^{-k} \end{aligned} \tag{13}$$

$$\begin{aligned} G_{0e}(z) &= \sum_k g_{0_{2k}}z^{-k} \ and\ G_{0o}(z) = \sum_k g_{0_{2k+1}}z^{-k} \\ G_{1e}(z) &= \sum_k g_{1_{2k}}z^{-k} \ and\ G_{1o}(z) = \sum_k g_{1_{2k+1}}z^{-k} \end{aligned} \tag{14}$$

$$\begin{aligned} F_{0e}(z) &= \sum_k f_{0_{2k}}z^{-k} \ and\ F_{0o}(z) = \sum_k f_{0_{2k+1}}z^{-k} \\ F_{1e}(z) &= \sum_k f_{1_{2k}}z^{-k} \ and\ F_{1o}(z) = \sum_k f_{1_{2k+1}}z^{-k} \end{aligned} \tag{15}$$

$$\begin{aligned} P_{0e}(z) &= \sum_k p_{0_{2k}}z^{-k} \ and\ P_{0o}(z) = \sum_k p_{0_{2k+1}}z^{-k} \\ P_{1e}(z) &= \sum_k p_{1_{2k}}z^{-k} \ and\ P_{1o}(z) = \sum_k p_{1_{2k+1}}z^{-k} \end{aligned} \tag{16}$$

We then assemble the polyphase matrices

$$P_{10}(z) = \begin{pmatrix} F_{0e}(z) & F_{1e}(z) \\ F_{0o}(z) & F_{1o}(z) \end{pmatrix} \tag{17}$$

$$P_{20}(z) = \begin{pmatrix} P_{0e}(z) & P_{1e}(z) \\ P_{0o}(z) & P_{1o}(z) \end{pmatrix} \tag{18}$$

So that

$$P_{1o}(z^2)^t = \frac{1}{2}M_{10}(z)\begin{bmatrix} 1 & z \\ 1 & -z \end{bmatrix} \tag{19}$$

$$P_{2o}(z^2)^t = \frac{1}{2}M_{20}(z)\begin{bmatrix} 1 & z \\ 1 & -z \end{bmatrix} \tag{20}$$

We now define $\widetilde{P_{10}(z)}$ and $\widetilde{P_{20}(z)}$ as dual version of $P_{10}(z)$ and $P_{20}(z)$,then the perfect reconstruction property yields that

$$P_{10}(z)\widetilde{P_{10}(z^{-1})}^t = \mathbf{I} \tag{21}$$

$$P_{20}(z)\widetilde{P_{20}(z^{-1})}^t = \mathbf{I} \tag{22}$$

© 2013 ACEEE
DOI: 01.IJSIP.4.3. 14

Where P(z) and $\widetilde{P(z)}$ contain only Laurent polynomials. The resultant real and imaginary wavelet transforms in polyphase notation can be represented schematically in figure (2)

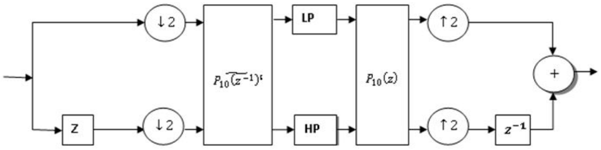


Fig 2.a:Polyphase representation of real tree wavelet transform.

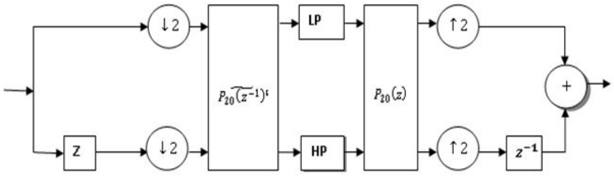


Fig 2b.Polyphase representation of imaginary tree wavelet transform.

In both cases first the image has been sub sampled into its even and odd parts, then apply the dual polyphase matrices. For inverse transform case, first apply the polyphase matrix and then join the even and odd parts. The major problem of an FIR wavelet transform resides in finding the matrices $P_{10}(z)$and $P_{20}(z)$ with determinants as one. Once we have such a matrices,$\widetilde{P_{10}(z)}$ and $\widetilde{P_{20}(z)}$ the four filters for the wavelet transform will be derived immediately from polyphase matrices and . From Cramer's rule

$$H_{0e}(z) = F_{1o}(z^{-1}) \ , \qquad H_{0o}(z) = -F_{1e}(z^{-1})$$

$$H_{1e}(z) = -F_{0o}(z^{-1}) \ and \ H_{1o}(z) = F_{0e}(z^{-1}) \tag{23}$$

$$G_{0e}(z) = P_{10}(z^{-1}) \ and \ G_{0o}(z) = -P_{1e}(z^{-1})$$

$$G_{1e}(z) = -P_{0o}(z^{-1}) \ and \ G_{1o}(z) = P_{0e}(z^{-1}) \tag{24}$$

This implies

$$H_1(z) = z^{-1}F_0(-z^{-1}) \ and \ H_0(z) = -z^{-1}F_1(-z^{-1}) \tag{25}$$

$$G_1(z) = z^{-1}P_0(-z^{-1}) and \ G_0(z) = -z^{-1}P_1(-z^{-1}) \tag{26}$$

The most trivial example of a polyphase matrix is P(z)=**I**. This will result in, $F_0(z) = H_0(z) = 1$ and

$F_1(z) = H_1(z) = z^{-1}$. Thus the wavelet transform does nothing else but sub sampling even and odd samples. This transform is generally called as a polyphase transform, but in the context of lifting it is often referred to as the lazy wavelet transform. This lazy wavelet transform [4] incorporates a numerous complex computations, which will consume a large amount of power, and time for computation, which results in high coding and computational complexity.

To avoid these hard striking shortcomings due to process complexity and delay, the natural wavelets are further configured to yield a simple, flexible and computationally efficient second generation wavelet structure..The robust lifting scheme [4] [5] was introduced to build second generation wavelets referred commonly as lifted wavelets, which are not necessarily translates and dilates of one fixed function [6].This construction is entirely spatial and therefore ideally suited for building second generation wavelets, that are more general in the sense that all the classical wavelets can be generated by the lifting scheme. For faster implementation of wavelet transform,the lifting scheme makes an optimal use of the similarities between the high pass and the low pass filters. The flexibility afforded by the lifting scheme allows the basis functions associated with wavelet coefficients near the window' boundaries to change their general shape .This new basis function can be used to minimize boundary effects. Being a new method for biorthogonal wavelet construction, the lifting scheme allows the faster implementation of wavelet transform. The number of flops can be reduced by a factor of two. The basic block diagram of lifting steps is illustrated in figure .3

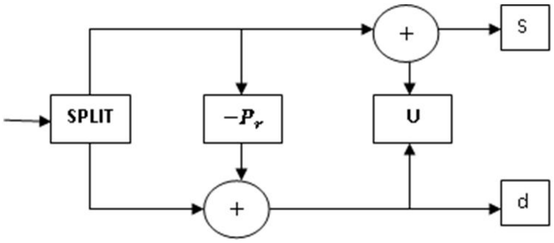


Fig. 3: Representation of lifting steps.

Where P_r is a predict[7] operator and **U** is an update operator[8].The lifting scheme can also be used in situations, such as wavelets on bounded domains, wavelets on curves and surfaces, weighted wavelets, and wavelets with irregular sampling. Conceptually the lifting scheme originates with a 'lazy wavelet', a function which essentially doesn't do a thing , but having a formal properties of a wavelet. The lifting scheme then generally builds a new wavelet, with improved properties by adding in, a several new basis functions to the lazy wavelet to make it faster as shown in figure(4). This is the inspiration behind the name 'Lifting Scheme'.

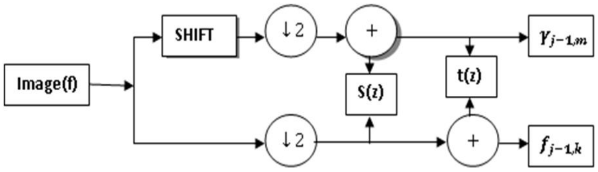


Fig.4: Fast wavelet transform using lifting scheme

which first does a lazy wavelet transform and then computes $\gamma_{j-1,m}$ and finally lifts the wavelet coefficients of the input image

The lifting scheme uses the relationship between the perfect reconstruction filter pairs$\{f_0(n), f_1(n)\}$,that have the same low pass or high pass filter. The filter pairs $\{f_0(n), f_1(n)\}$ and $\{p_0(n), p_1(n)\}$ to be complementary if in case the corresponding polyphase matrices $P_{10}(z)$ and $P_{20}(z)$ have their determinant as 1. and if the filter pair,{} and {} are complementary, then consequently the filter pairs {} and {}are also complementary. Then any other finite filters and, complementary to and respectively are of the form

$$F_1^{new}(z) = F_1(z) + F_0(z)S(z^2) \tag{27}$$

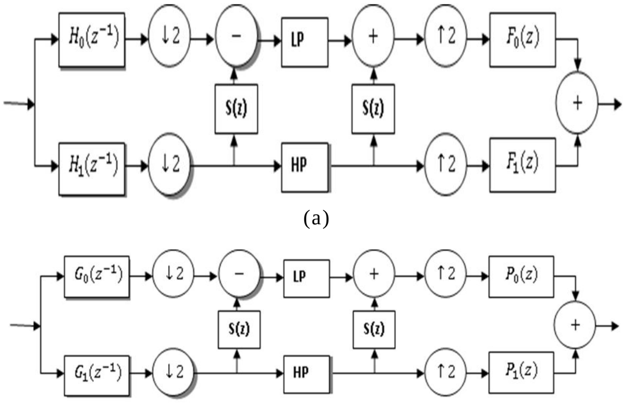
$$P_1^{new}(z) = P_1(z) + P_0(z)S(z^2) \tag{28}$$

Where S(z) is a Laurent polynomial. Similarly as per as dual lifting concerned, there exists any finite filters of the form,

$$F_0{}^{new}(z) = F_0(z) + F_1(z)t(z^2) \quad (29)$$

$$F_0{}^{new}(z) = F_0(z) + F_1(z)t(z^2) \quad (30)$$

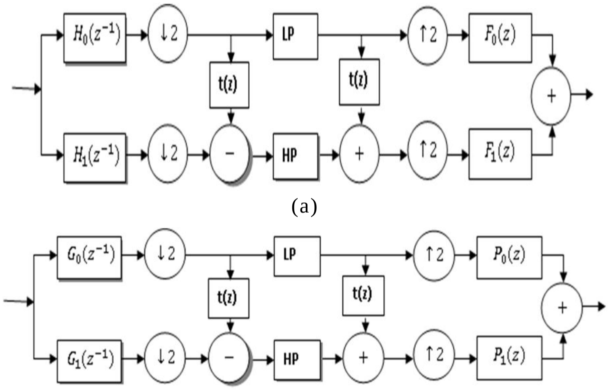
Where **t(z)** is a Laurent polynomial. The resulted lifted wavelet tree structure for the case of classical sub band scheme ,when lifted the lowpass sub band with the help of the high pass sub band for both real and imaginary trees is shown in figure(5).And in a similar way the dual lifted wavelet tree structure for the case of classical sub band scheme, when lifted the high pass sub band with the help of the low pass sub band for both real and imaginary trees is shown in figure(6).



(a)

(b)

Fig .5.The Lifting scheme (a) real (b) imaginary trees: first a classical sub band filters scheme and then lifting the low pass sub band with the help of the high pass sub band.



(a)

(b)

Fig. 6:The dual lifting scheme (a) real,(b) imaginary wavelet trees: First a classical sub band filter scheme and later lifting the high pass sub band with the help of the low pass sub band.

III. Proposed Work

In this paper, it is proposed to reduce the computational complexity, computational time, power and complexity of coding structure, with a substantial growth in average computational speed, and coding gain in a traditional dual tree complex wavelet transform, which employs two separate Quadrature distinct discrete wavelet structures in which one is taken as the real tree wavelet structure[9][10],and the other is taken as the imaginary tree wavelet structure. Each discrete wavelet transform structure irrespective of its nature will have a pair of low pass and high pass analysis filters and a similar pair of low pass and high pass synthesis filters. The analysis and synthesis filter pair in one tree will differ with those in other tree by an half sample delay. But the wavelets in the traditional system are lazy wavelets(i.e., computationally complex, slower and consumes a high quantity of power).Hence to modernize the wavelets and to cater the hard striking inabilities of traditional wavelets. The traditional wavelets are further processed and modified, to lift them, in a view to get a more simple, flexible, linear, complexity free and fast second generation wavelets[11]. These second generation wavelets are constructed, first by lifting the lazy wavelet which involves bi orthogonal factorization [12][13] [14] [15] of lazy wavelet filters to eradicate delay artifacts and to remove the coefficients in the filter transfer functions responsible for delayed processing, or operational complexity, consuming higher processing power, to yield the lifted filters for second generation wavelets, called lifted wavelet filters for computationally faster lifted wavelet transform.

The process of the lifting wavelet construction involves factorization of wavelet filters to get simple lifted wavelet filters. This activity involves running Euclidean algorithm starting from polyphase components of $F_0(z)$ and $P_0(z)$ as $\{F_{0e}(z), F_{0o}(z)\}$ and $\{P_{0e}(z), P_{0o}(z)\}$ respectively and their gcd's are monomials. With the non-uniqueness of the division, we can always choose the quotients $q_i(z)$,so that the gcd is a constant. Let this constant be K_n then we thus have

$$\begin{bmatrix} F_{oe}(z) \\ F_{oo}(z) \end{bmatrix} = \prod_{i=1}^{n} \begin{bmatrix} q_i(z) & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} K_n \\ 0 \end{bmatrix} \quad (31)$$

$$\begin{bmatrix} P_{oe}(z) \\ P_{oo}(z) \end{bmatrix} = \prod_{i=1}^{n} \begin{bmatrix} q_i(z) & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} K_n \\ 0 \end{bmatrix} \quad (32)$$

in case $|F_{0o}(z)| > |F_{0e}(z)|$,the first quotient $q_1(t)$ is zero. We can always assume that 'n' is even. If we know the filters $F_0(z)$ and $P_0(z)$,then we can always find the complementary filters$F_1{}^0(z)$ and $P_1{}^0(z)$ by letting

$$P_{10}{}^0(z) = \begin{bmatrix} F_{0e}(z) & F_{1e}{}^0(z) \\ F_{0o}(z) & F_{1o}{}^0(z) \end{bmatrix} \prod_{i=1}^{n} \begin{bmatrix} q_i(z) & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} K_n & 0 \\ 0 & \frac{1}{K_n} \end{bmatrix} \quad (33)$$

$$P_{10}{}^0(z) = \begin{bmatrix} P_{0e}(z) & P_{1e}{}^0(z) \\ P_{0o}(z) & P_{1o}{}^0(z) \end{bmatrix} \prod_{i=1}^{n} \begin{bmatrix} q_i(z) & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} K_n & 0 \\ 0 & \frac{1}{K_n} \end{bmatrix} \quad (34)$$

Here the final diagonal matrix follows from the fact that det ($P_{10}{}^0(z)$)=det($P_{20}{}^0(z)$)=1 and 'n' is even, then by rewriting the above equations, we observe that

$$\begin{bmatrix} q_i(z) & 1 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} 1 & q_i(z) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ q_i(z) & 1 \end{bmatrix} \quad (35)$$

Using the first equation of above in case 'i' is odd and the second in case is even yields

$$P_{10}{}^0(z) = \prod_{i=1}^{n/2} \begin{bmatrix} 1 & q_{2i-1}(z) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ q_i(z) & 1 \end{bmatrix} \begin{bmatrix} K_n & 0 \\ 0 & \frac{1}{K_n} \end{bmatrix} \quad (36)$$

As an example to this factorization process to built computationally speed lifting wavelets, let us consider a lazy

© 2013 ACEEE
DOI: 01.IJSIP.4.3. 14

wavelet of haar type. Now our task is to factor the filters associated with lazy haar wavelet transform to build ,active lifting wavelet filters there by to build lifted haar wavelets. In the case of unnormalized wavelets we know that for a single tree

$$F_0(z) = 1 + z^{-1}\ , F_1(z) = \frac{-1}{2} + \frac{1}{2}z^{-1},$$
$$H_0(z) = \frac{1}{2} + \frac{1}{2}z^{-1}, H_1(z) = -1 + z^{-1}$$

using the aforementioned Euclidean algorithm, we can thus write the polyphase matrices as

$$P_{10}(z) = \begin{bmatrix} 1 & \frac{-1}{2} \\ 1 & \frac{1}{2} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} 1 & \frac{-1}{2} \\ 0 & 1 \end{bmatrix} \quad (37)$$

$$P_{20}(z) = \begin{bmatrix} 1 & \frac{-1}{2} \\ 1 & \frac{1}{2} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} 1 & \frac{-1}{2} \\ 0 & 1 \end{bmatrix} \quad (38)$$

Thus on the analysis side we have

$$P_{10}(z)^{-1} = P_{11}\left(\frac{1}{z}\right) = \begin{bmatrix} 1 & \frac{1}{2} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ -1 & 1 \end{bmatrix} \quad (39)$$

$$P_{20}(z)^{-1} = P_{21}\left(\frac{1}{z}\right) = \begin{bmatrix} 1 & 1/2 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ -1 & 1 \end{bmatrix} \quad (40)$$

Thus the forward and reverse wavelet transform implementation structure using the alternating lifting and dual lifting steps are better illustrated in figure.7

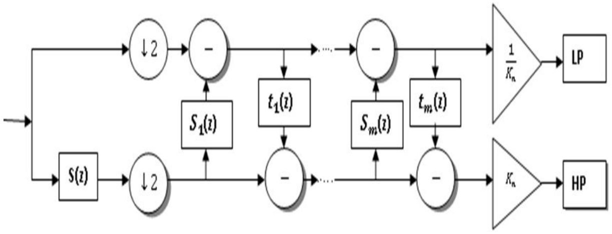


Fig .7a .Forward wavelet transform using the lifting scheme.

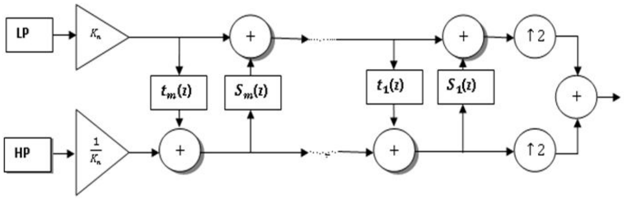


Fig.7.b:The inverse wavelet transform using the lifting scheme

$$S_l{}^0 = x_{2l} \qquad d_l{}^0 = x_{2l+1}$$
$$d_l = d_l{}^0 - S_l{}^0 \qquad S_l = S_l{}^0 + \frac{1}{2}d_l \quad (41)$$

Similarly the inverse factorization yields,

$$S_l{}^0 = S_l - \frac{1}{2}d_l \quad , \quad d_l{}^0 = d_l + S_l{}^0,$$
$$x_{2l+1} = d_l{}^0 \quad , \quad x_{2l} = S_l{}^0 \quad (42)$$

Thus the lifting wavelets are designed for both real and imaginary trees ,and incorporated to get real and imaginary discrete wavelet transform[11] trees, which are suitably combined to form the dual tree complex wavelet transform to erase the limitations of the single tree discrete wavelet transform, such as shift variance, poor directionality, loss of phase information and aliasing.

Thus for the case of a boir 6.8 wavelet type, the corresponding lifting scheme LS = {...

```
 'p'    [ -2.65899636  0.99715069]              [0]
 'd'    [  0.27351197  0.27351197]              [0]
 'p'    [  3.87782215 -3.26868661]              [2]
 'd'    [ -0.28650326 -0.28650326]             [-2]
 'p'    [ -0.54859417  2.94176754]              [4]
 'd' [ 0.09982322 -0.34381326 -0.34381326 0.09982322]
  [                                            [-2]
  [ 0.86857870] [ 1.15130615]                  [73]
};
```

The transfer functions of the corresponding lifted wavelet analysis filters for the dual tree complex wavelet transform are given in figure (8)

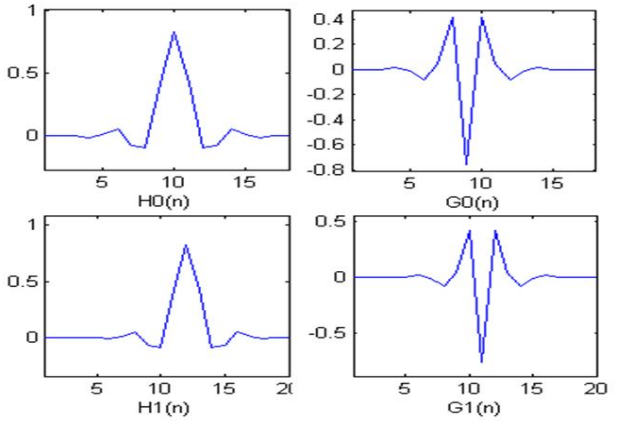


Fig .8:Transfer functions of lifted wavelet dual tree analysis filters.

Similarly the transfer functions of the corresponding lifted wavelet synthesis filters for the dual tree complex wavelet transform are given in figure .9

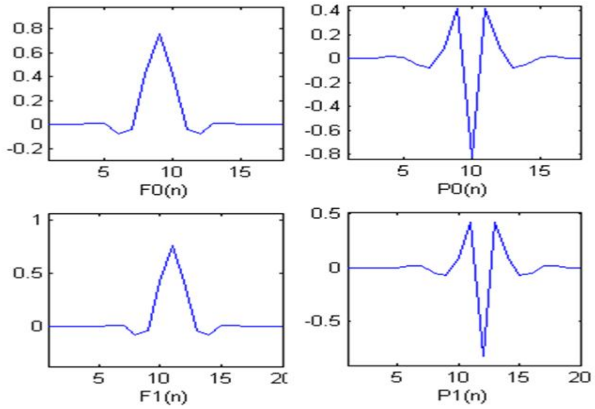


Fig. 9:Transfer functions of lifted wavelet dual tree synthesis filters.

The corresponding lifted wavelet coefficients are given for the dual tree complex lifting wavelet transform both for analysis and synthesis separately as in table(1) and table (2).

Thus a computationally fast dual tree complex wavelet transform is constructed based on lifted wavelets .Now this efficient and faster DTCWT tool is employed as a tool of interest for an image denoising. In this paper a noise degraded image is first applied to both real and imaginary discrete wavelet transform trees to obtain real and imaginary coefficients. These coefficients are suitably combined to get complex wavelet coefficients.

The corresponding dual tree wavelet structure is shown in figure.10

© 2013 ACEEE
DOI: 01.IJSIP.4.3.14

TABLE I. ANALYSIS LIFTED WAVELET COEFFICIENTS FOR DTCWT WITH LS.

$H_0(n)$	$H_1(n)$	$G_0(n)$	$G_1(n)$
0		0	0
0.0019	0	0	0
-0.0019	0	0	0
-0.0170	0	0.0019	0
0.0119	0.0144	-0.0019	0
0.0497	-0.0145	-0.0170	0.0144
-0.0773	-0.0787	0.0119	-0.0145
-0.0941	0.0404	0.0497	-0.0787
0.4208	0.4178	-0.0773	0.0404
0.8259	-0.7589	-0.0941	0.4178
0.4208	0.4178	0.4208	-0.7589
-0.0941	0.0404	0.8259	0.4178
-0.0773	-0.0787	0.4208	0.0404
0.0497	-0.0145	-0.0941	-0.0787
0.0119	0.0144	-0.0773	-0.0145
-0.0170	0	0.0497	0.0144
-0.0019	0	0.0119	0
0.0019	0	-0.0170	0
	0	-0.0019	0
		0.0019	0

TABLE II. SYNTHESIS LIFTED WAVELET COEFFICIENTS FOR DTCWT WITH LS

c	$F_1(n)$	$P_0(n)$	$P_1(n)$
		0	0
0	0	0	0
0	-0.0019	0	0
0	-0.0019	0	-0.0019
0.0144	0.0170	0	-0.0019
0.0145	0.0119	0.0144	0.0170
-0.0787	-0.0497	0.0145	0.0119
-0.0404	-0.0773	-0.0787	-0.0497
0.4178	0.0941	-0.0404	-0.0773
0.7589	0.4208	0.4178	0.0941
0.4178	-0.8259	0.7589	0.4208
-0.0404	0.4208	0.4178	-0.8259
-0.0787	0.0941	-0.0404	0.4208
0.0145	-0.0773	-0.0787	0.0941
0.0144	-0.0497	0.0145	-0.0773
0	0.0119	0.0144	-0.0497
0	0.0170	0	0.0119
0	-0.0019	0	0.0170
0	-0.0019	0	-0.0019
		0	-0.0019

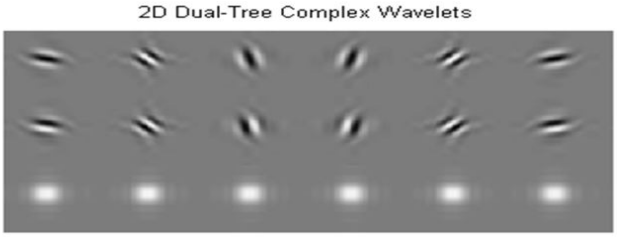


Fig.10:Dual tree complex lifted wavelet structure.

The dual tree complex wavelet transform embedded with lifting scheme is initially aimed to achieve a noteworthy progress in the computational speed up in the image analysis. This computationally outstanding tool integrated with the globally accepted adapted wavelet threshoulding techniques have been employed as the key techniques to develop a modified denoising algorithm for image denoising. Since the dual tree complex wavelet transform with lifting structure incredibly accelerate the decomposition process with the conserved shift invariance and directional selective properties as earlier , adaptive soft thresholding technique central to the adaptive wavelet threshoulding has been integrated to denoise the image. The algorithm is intelligibly presented and implemented as

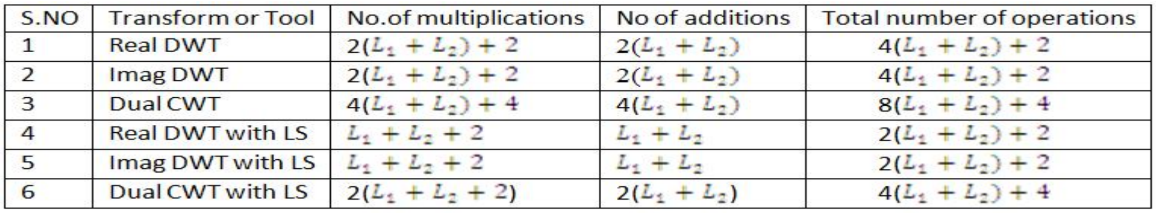
1. Select the test image of size 256X256.
2. Select the type of the wavelet to be lifted.
3. Lift the selected wavelet with sufficient predict and update operations to generate the lifted wavelet structure of the selected wavelet .
4. Design and generate the lifted wavelet filters associated with the lifted wavelet(Both for analysis and synthesis).
5. Apply the dual tree based complex wavelet transform on the test image using the lifted wavelet filter pair for analysis.
6. Grouping the connected components using the neighborhood based statistical modeling of the complex wavelet coefficients.
7. Apply the adaptive soft thresholding technique with a threshold value of 40.
8. Applly the inverse dual tree complex wavelet transform on the thresholded complex wavelet coefficients using the lifted wavelet filters for synthesis.
9. Compare the results obtained in terms of PSNR and RMSE.

A. Computational complexity

In this section ,we used to focus on the computational complexity of the wavelet transform .As a comparison base we used the standard algorithm(DTCWT), which corresponds to applying the polyphase matrix. This will take the advantage of the fact that the filters will be sub sampled and thus avoids computing samples that will be sub sampled immediately. For each filter either it may be low pass or high pass,the number of multiplications and additions required are $|H_0(z)| + 1$ and $|H_0(z)|$ respectively.Thus in a standard algorithm without applying lifting scheme, the number of multiplications and additions required for a single discrete wavelet transform tree are $(|H_0(z)| + |H_1(z)| + 2)$ and $((|H_0(z)| + |H_1(z)|)$ respectively. Similarly for an imaginary tree, the $(|G_0(z)| + |G_1(z)| + 2)$number of multiplications and$(|G_0(z)| + |G_1(z)|)$ number of additions are required. Thus for dual tree complex wavelet transform, the total number of multiplications and additions needed are$(|H_0(z)| + |H_1(z)| + |G_0(z)| + |G_1(z)| + 4)$ and $(|H_0(z)| + |H_1(z)| + |G_0(z)| + |G_1(z)|)$ respectively. For the case of symmetric filters let $|H_0(z)| = 2L_1 = |G_0(z)|$, $|H_1(z)| = 2L_2 = |G_1(z)|$, and also be assumed that $L_2 \geq L_1$.Then for a standard algorithm ,the number of multiplications and additions required for a real tree are 2L_1+2L_2+2 and 2$L_1 + 2L_2$.Similarly for an imaginary tree the total number of multiplications and additions required are 2+2+2 and 2.Then for the dual tree complex wavelet transform, the total number of multiplications and additions required are 4(4 and 4() respectively. Similarly if the lifting scheme is incorporated, then for a single tree, 2 scaling operations are required, for lifting steps,4 lifting operations are required and for final lifting steps, the number of operations required are 2(+1).Then the total number of operations required are 2(),among which the total number of multiplications and additions are and respectively. Thus for the dual tree complex wavelet transform using the lifting scheme the total number of 2() multiplications and 2(additions are required. Correspondingly the computational operations required for various transform cases are investigated in table.3.

Graphically the computational complexity of the proposed algorithm with DTCWT-LS is compared with DTCWT as shown in fig.11. As computational complexity increases, the computational time increases and vice-versa. Thus an algorithm with higher computational complexity will take higher time for its computation and that with lower computational complexity will take less time for its computation on the same processor. Thus the computational time with several wavelets are compared in terms of no of computations required as in table .4

TABLE III. COMPARISON OF COMPUTATIONAL COMPLEXITY BETWEEN STANDARD ALGORITHM AND LIFTING ALGORITHM.



S.NO	Transform or Tool	No.of multiplications	No of additions	Total number of operations
1	Real DWT	$2(L_1+L_2)+2$	$2(L_1+L_2)$	$4(L_1+L_2)+2$
2	Imag DWT	$2(L_1+L_2)+2$	$2(L_1+L_2)$	$4(L_1+L_2)+2$
3	Dual CWT	$4(L_1+L_2)+4$	$4(L_1+L_2)$	$8(L_1+L_2)+4$
4	Real DWT with LS	L_1+L_2+2	L_1+L_2	$2(L_1+L_2)+2$
5	Imag DWT with LS	L_1+L_2+2	L_1+L_2	$2(L_1+L_2)+2$
6	Dual CWT with LS	$2(L_1+L_2+2)$	$2(L_1+L_2)$	$4(L_1+L_2)+4$

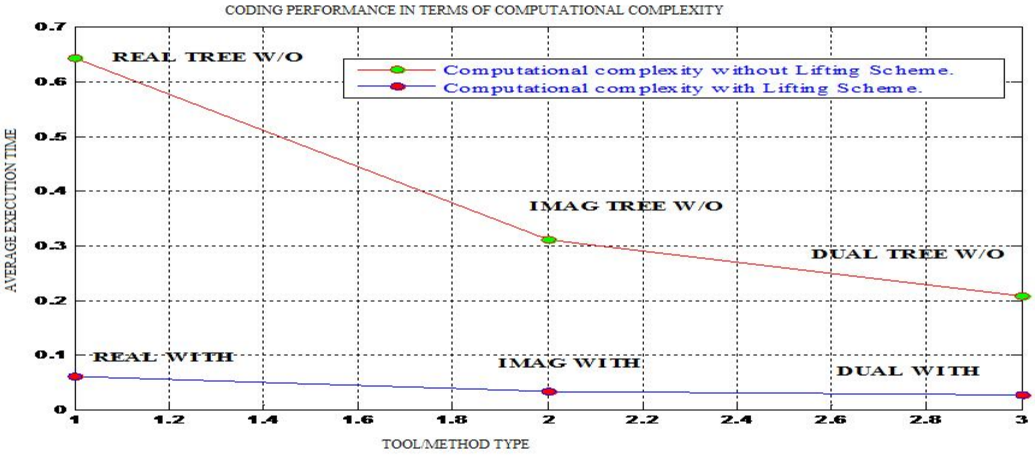
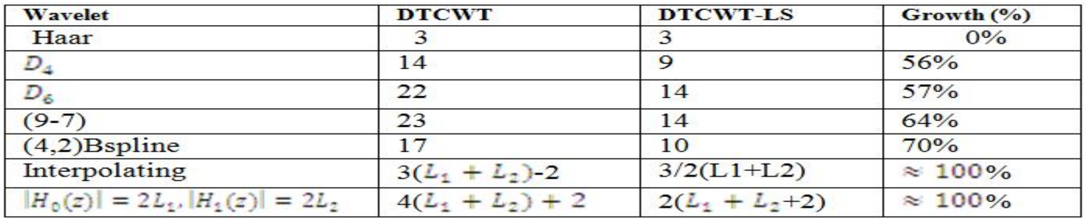


Fig.11

TABLE IV COMPARISON OF COMPUTATIONAL TIME BETWEEN STANDARD ALGORITHM AND LIFTING ALGORITHM FOR VARIOUS WAVELET FILTERS

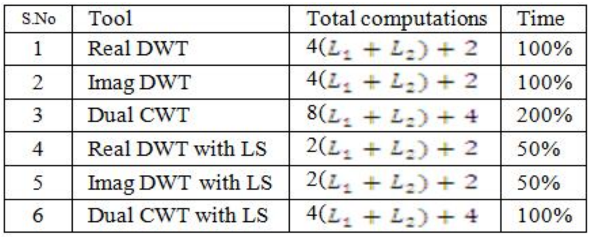


Wavelet	DTCWT	DTCWT-LS	Growth (%)
Haar	3	3	0%
D_4	14	9	56%
D_6	22	14	57%
(9-7)	23	14	64%
(4,2)Bspline	17	10	70%
Interpolating	$3(L_1+L_2)$-2	3/2(L1+L2)	≈ 100%
$\lvert H_0(z)\rvert = 2L_1, \lvert H_1(z)\rvert = 2L_2$	$4(L_1+L_2)+2$	$2(L_1+L_2$+2)	≈ 100%

B. Computational Time:

As the number of operations are increases then the computational time will also increases. Thus the computational time of a standard and lifting algorithms are compared as in table.5,.Inversely the computational time of the dual tree complex wavelet transform with lifting scheme will be compared graphically with that of DTCWTas shown in figure.12.

TABLE V



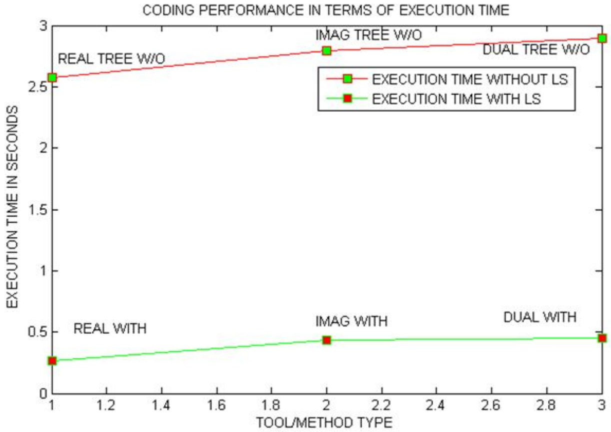
S.No	Tool	Total computations	Time
1	Real DWT	$4(L_1+L_2)+2$	100%
2	Imag DWT	$4(L_1+L_2)+2$	100%
3	Dual CWT	$8(L_1+L_2)+4$	200%
4	Real DWT with LS	$2(L_1+L_2)+2$	50%
5	Imag DWT with LS	$2(L_1+L_2)+2$	50%
6	Dual CWT with LS	$4(L_1+L_2)+4$	100%

C. Coding gain:

As the number of operations in the lifting scheme based real ,imaginary and dual tree complex wavelet transform are exactly half of those in standard real, imaginary and dual tree complex wavelet transforms respectively, then the overall coding gain will increases by 100%.Graphically the coding

© 2013 ACEEE
DOI: 01.IJSIP.4.3.14

gain of the DTCWT-LS is compared with that of the DTCWT as shown in figure 13



Fig(12):Compaison of computational time of DTCWT-LS with that of the DTCWT

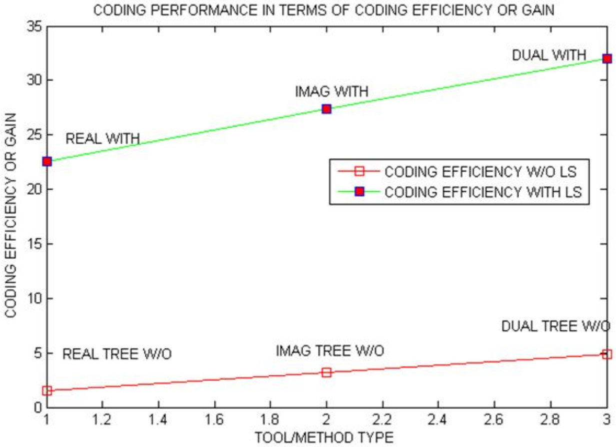


Figure 13

IV. Results and Discussion

The denoising results are presented and compared with the former algorithms employing the conventional DTCWT and adaptive wavelet thresholding. The simulation results obtained have ascertained that there is a remarkable progress in the PSNR value and notable drop in the RMSE value of the algorithm compared to the former algorithm. Practically a PSNR value of 40.04 and an RMSE value of 05.0267 are possible for additive white Gaussian noise(AWGN),but with the former algorithm the maximum PSNR value of 22.8709 and minimum RMSE value of 14.9928 are only possible with AWGN. With an impulse noise ,the modified algorithm offers a maximum PSNR of 47.4560 and minimum RMSE of 04.5656,where as the former algorithm can only offer a maximum PSNR of 24.2356 and a minimum RMSE of 12.5612..An important note here is that all the results are obtained with a Barbara.jpg image of 256X256 as a test image.

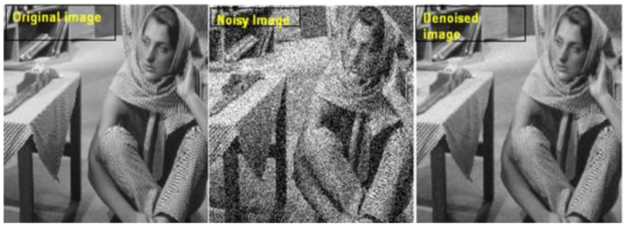
A comparative analysis of the performance offered by the DTCWT without lifting scheme(LS) and the DTCWT with lifting scheme are done in terms of the no of computations required and the amount of time elapsed to perform them for a particular image format. The performance of the both the algorithms have been compared with several images of different size and format and the results are given in table.7 .For instance with the DTCWT without LS algorithm the no of complex computations required to process a cameraman image of size 256X256 and of tif format, a real tree requires 9568256 computations ,for an imaginary tree 9568256 computations are needed and for dual tree on a whole 19136512 computations are essential. But with DTCWT with LS algorithm ,to process the same image it requires 4849664 real, 4849664 imaginary and 9699328 dual tree computations are sufficient. Similarly to process a Barbara image of jpg format and of 256X256 size the DTCWT without LS algorithms takes a time of 0.1486 seconds for real tree, 0.024 seconds for imaginary tree and on a whole 0.2106 seconds are essential for dual tree. Where as to process the same image the DTCWT with LS consumes a time of 0.1239 seconds for real tree,0.0188 seconds for imaginary tree and on a total 0.1436 seconds for dual tree is enough. The testing results have strongly witnessed the computational dominance of the DTCWT with LS algorithm over the DTCWT without LS algorithm, and outstandingly judged the computational efficiency of the DTCWT with LS algorithm In order to compare the quality of the denoised image signals obtained with both the algorithms such as DTCWT without lifting scheme and DTCWT with lifting scheme, the percent root mean square formula for both original and reconstructed image signals is

$$PRD = \frac{\sqrt{\Sigma(x_i - x_j)^2}}{\sqrt{\sum x_j^2}} X100 \qquad (43)$$

Where x_i is the denoised image signal obtained with the DTCWT without LS and x_j is the resulting denoised image signal obtained by the DTCWT. Similarly the outputs obtained with the DTCWT with LS and DTCWT without LS are here subtracted from the original input and differences are computed as RMS error (RMSE).The generalized formula for RMSE is

$$RMSE = \sqrt{\frac{1}{N}\sum_{1}^{N}(x_i - \widehat{x_i})^2} \qquad (44)$$

where $\widehat{x_i}$ is denoised image signal and x_i is the original image signal without any noise.



(a) (b) (c)

Fig.14:Visual Results obtained with Gaussian noise and (a).Original image(b)Noisy image (c)Denoised image.

The denoising results obtained with Gaussian noise and impulse noise are illustrated in figure.14 & 15.

Image quality performance metrics PSNR and MSE are computed and are tabulated as given in below table.6.

© 2013 ACEEE
DOI: 01.IJSIP.4.3.14

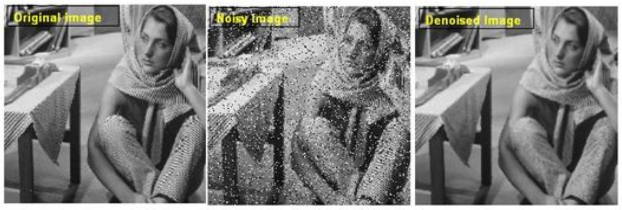


Fig.15Visual Results obtained with an Impulse noise (a).Original image, (b)Noisy image (c)Denoised image.

TABLE VI

Noise Type	Technique	PSNR	RMSE
AWGN	DTCWT	32.0637	6.3585
	DTCWT-LS	47.8222	15.5608
Impulse	DTCWT	22.8946	18.2731
	DTCWT-LS	48.4823	12.9214

V. CONCLUSIONS

In this paper we developed a algorithm which suitably combines the salient features of nearly shift invariant, directionally selective dyadic decomposition tree based dual tree complex wavelet transforms deploying quadrature distinct ,perfect reconstruction filter banks and linear adaptive, and faster decomposition arithmetic based robust lifting scheme for an image denoising in a view to get a substantial speed up in the decomposition and reconstruction structure of the dual tree complex wavelet transform while preserving all performance salient features that results from the dual tree complex wavelet transform itself. This algorithm provides a substantial drop in the average decomposition time, average power required for decomposition as a major distinction factor from an ordinary dual tree complex wavelet decomposition algorithm. With this lifted Dual tree structure we achieved an accountable growth in the coding efficiency and coding gain. This algorithm first performs the factorization of the filters associated with the selected wavelet type based on the Euclidean algorithm and then constructs its associated lifting scheme from which the filters associated with the lifted wavelet will be derived. These filters are employed to achieve an in-place-implementation of the fast Dual tree Wavelet structure. This computationally efficient, dual tree complex wavelet structure has been employed as a tool of interest for an image denoising and achieved a considerable growth in image quality and performance metrics viz., PSNR and RMSE.

REFERENCES

[1] M. G. Bellangerand J. L. Daguet. TDM-FDM transmultiplexer: Digital polyphase and FFT. IEEE Trans. Commun., 22(9): 1199.1204, 1974

[2] S.G.Mallat.Multifrequency channel decompositions of images and wavelet models. IEEE Trans. Acoust. Speech Signal Process., 37(12):2091.2110, 1989.

[3] C. Herley and M. Vetterli. Wavelets and recursive filter banks. IEEE Trans. Signal Process., 41(8):2536.2556, 1993

[4] W. Sweldens. The lifting scheme: A custom-design construction of biorthogonal wavelets. Appl. Comput. Harmon. Anal.,3(2):186.200, 1996

[5] W. Sweldens. The lifting scheme: A construction of second generation wavelets. SIAM J. Math. Anal., 29(2):511.546,1997

[6] A. Grossmann and J. Morlet. Decompostion of Hardy functions into square integrable wavelets of constant shape. SIAM J. Math. Anal., 15(4):723.736, 1984

[7] A. Harten. Multiresolution representation of data: A general framework. SIAM J. Numer. Anal., 33(3):1205.1256, 1996.

[8] W. Sweldens and P. Schr¨oder. Building your own wavelets at home. In Wavelets in Computer Graphics, pages 15.87.ACM SIGGRAPH Course notes, 1996.

[9] T. Q. Nguyen and P. P. Vaidyanathan. Two-channel perfect-reconstruction FIR QMF structures which yield linear-phase analysis and synthesis _lters. IEEE Trans. Acoust. Speech Signal Process., 37:676.690, 1989

[10] P. P. Vaidyanathan and P.-Q. Hoang. Lattice structures for optimal design and robust implementation of two-band perfect reconstruction QMF banks. IEEE Trans. Acoust. Speech Signal Process., 36:81.94, 1988

[11] G. Strang and T. Nguyen. Wavelets and Filter Banks. Wellesley, Cambridge, 1996..

[12] T. A. C. M. Kalker and I. Shah. Ladder Structures for multidimensional linear phase perfect reconstruction filter banks and wavelets. In Proceedings of the SPIE Conference on Visual Communications and Image Processing (Boston), pages 12.20, 1992

[13] L. M. G. Tolhuizen, H. D. L. Hollmann, and T. A. C. M. Kalker. On the realizability of bi-orthogonal M- dimensional 2-band filter banks. IEEE Transactions on Signal processing, 1995.

[14] T. G. Marshall. A fast wavelet transform based upon the Euclidean algorithm. In Conference on Information Science and Systems, Johns Hopkins, MD, 1993.

[15]. T. G. Marshall. U-L block-triangular matrix and ladder realizations of subband coders. In Proc. IEEE ICASSP, volume III, pages 177.180, 1993.

TABLE VII TESTED RESULTS FOR COMPUTATIONAL COMPLEXITY

Image	No of Operations						Computation time					
	Dual Tree CWT			For Lifting scheme DTCWT			Dual Tree CWT			DTCWT-LS		
	Real	Imaginary	Dual	Real	Imaginary	Dual	Real	Imaginary	Dual	Real	Imaginary	Dual
cameraman	9568256	9568256	19136512	4849664	4849664	9699328	0.547	0.547	0.547	0.1293	0.0202	0.1562
peppers	28704768	28704768	57409536	14548992	14548992	29097984	0.1712	0.044	0.3051	0.1051	0.0309	0.1203
Lena	5840000	5840000	11680000	2960000	2960000	5920000	0.1455	0.0194	0.1951	0.1212	0.0162	0.1629
Brandyrose	8380400	8380400	16760800	4247600	4247600	8495200	0.1515	0.0217	0.2073	0.1241	0.0175	0.1402
Barbara	9568256	9568256	19136512	4849664	4849664	9699328	0.1486	0.024	0.2106	0.1239	0.0188	0.1436
Bear	8906000	8906000	17812000	4514000	4514000	9028000	0.1536	0.0229	0.212	0.1233	0.0182	0.1313
Kid	8438800	8438800	16877600	4277200	4277200	8554400	0.1455	0.0219	0.2047	0.1229	0.0177	0.1763
Watch	4380000	4380000	8760000	2220000	2220000	4440000	0.1454	0.0189	0.1928	0.1205	0.0154	0.1564
Parrot	9568256	9568256	19136512	4849664	4849664	9699328	0.1479	0.0236	0.2088	0.125	0.0201	0.1451
Baboon	5840000	5840000	11680000	2960000	2960000	5920000	0.1442	0.0201	0.1942	0.103	0.0162	0.1192
House	9568256	9568256	19136512	4849664	4849664	9699328	0.1582	0.0247	0.2199	0.1254	0.0196	0.1396

© 2013 ACEEE
DOI: 01.IJSIP.4.3. 14